Module 2
Data Types and Expressions

Tamer Abdou, PhD

IND 830 - Python Programming for Data Science

Data Science Laboratory

Ryerson University

Ryerson University
The Chang School of Continuing Education

Outline

1. Basic Types and Text Processing
2. Numeric Data Types
3. Functions and Modules

Ryerson University
The Chang School of Continuing Education

Basic Types and Text Processing

1 Basic Types and Text Processing
- Strings
- Variables
- Comments

2 Numeric Data Types

3 Functions and Modules

Types of Data

- **Data Type:** An attribute of data or a set of operations that can be performed on those values
- **Literal:** The way a value of a data type looks to a programmer, or the data value that appears in a program
- **Variable:** A name that refers to a specific value

Basic Data Types

Table 2.1: Data Types and Literals

Data Type	Python Code Name	Literal Example
Strings	str	'Hello', "Python", "3.78"
Integers	int	9, -7, 3, 6
Real Numbers	float	9.0, -7.21, 3.78
Booleans	bool	True, False
List	list	[1, 2, 4, 4, 5]
Set	set	{1, 2, 3, 4, 5}
Tuple	tuple	(1, 2, 3, 4, 5)
Dictionary	dict	{"one":1, "two":2}

Strings

- **String Literals:** A string literal is a sequence of characters enclosed in single or double quotation marks
- **Escape Sequences:** Python expresses special characters, such as the tab, newline, and backspace (delete key) as literals using backslash
- **String Concatenation:** Python uses the '+' operator to join two or more strings and form a new string

Variables and Assignment Statement

- Variables receive their initial values and can be reset to new values with an **assignment statement (=)**
- Some names, such as **if, def, and import**, are reserved for other purposes and thus cannot be used for variable names
- A variable name must begin with either a letter or an underscore (_), and can contain any number of letters, digits, or other underscores
- For variable names that consist of more than one word, it is common to begin each word in the variable name (except for the first one) with an uppercase letter (camelCase)

Examples of Variables

Table 2.2: Valid and Invalid Variable Names

Valid Name	Invalid Name	Error Message
cityName	city Name	SyntaxError: invalid syntax
StudentID2	2StudentID	SyntaxError: invalid syntax
x514	514x	SyntaxError: invalid syntax
True_Mean	True	SyntaxError: can't assign to keyword
TimeDate	Time&Date	SyntaxError: can't assign to operator
employee_salary	employee.salary	NameError: name 'employee' is not defined
_oddNumber	!oddNumber	/bin/bash: oddNumber: command not found

Ryerson University The Chang School of Continuing Education

Comments and Docstrings

- **Comment:** A piece of text which is ignored by the program interpreter
- **Docstring:** A style of commenting where a multi-line string is introduced at the beginning of the code and includes:
  1. The function/class name
  2. The name of the author of the program
  3. The creation/modification date
  4. A brief about the purpose of this function/class

Numeric Data Types

1 Basic Types and Text Processing

2 Numeric Data Types
- Integers
- Floating-Point Numbers
- Character Sets
- Arithmetic Expressions

3 Functions and Modules

Integers and Long Integers

- **int** data type consists of the integers from -2,147,483,648 (-2^{31}) to 2,147,483,647 ($2^{31} - 1$)
- If you are using a 64 bit operating system, the min value of an integer will be -2^{63} and the max value will be $2^{63} - 1$
- When the value of an integer exceeds these limits, Python automatically uses the **long** data type to represent it
- The long integer is a regular integer but ends with the letter **L**
- In Python 3.x, using the `type()` function will return `<class 'int'>` with no regard to the integer size.

Floating-Point Numbers

- **float** data type consists of:
  1. A whole integer number
  2. A decimal point
  3. A fractional part
- **Infinite precision** means that the digits in the fractional part can continue forever
- **Scientific notation** is often useful for mentioning very large numbers (e.g. $2.54e-8$)
- According to the ▸ IEEE-754 Standard, the maximum value a floating-point number can have is approximately $1.8e+308$

Character Set Map

Table 2.3: ASCII Character Set Map [Lambert, 2010]

	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
1	LF	VT	FF	CR	SO	SI	DLE	DCI	DC2	DC3
2	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
3	RS	US	SP	!	“	#	$	%	&	`
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	‘	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	DEL		

Precedence Rules

Table 2.4: Precedence Rules of Arithmetic Operators

Rank	Operator	Syntax	Meaning
1	**	Exponentiation	$x**y$
2	*	Multiplication	$x*y$
2	/	Division	x/y
2	%	Modulus	$x\%y$
3	+	Addition	$x+y$
3	−	Subtraction	$x-y$

These rules come with two exceptions:

1. Operations of equal precedence are **left associative**
2. Exponentiation operations are **right associative**

Examples

Table 2.5: Expression and Corresponding Code

Mathematical Expression	Python Code
$4x + 3$	4 * x + 3
$\sqrt{\frac{x}{2}}$	(x/2)**(1/2)
$\frac{x+1}{1-y}$	(x+1)/(1-y)
$\frac{x^2+1}{1-y^3}$	(x**2+1)/(1-y**3)

Functions and Modules

1 Basic Types and Text Processing

2 Numeric Data Types

3 Functions and Modules
Calling Functions
Modules in Python

Functions

- A **function** is a piece of code that performs a specific task and can be reused to ensure maintainability
- Each function has a name and a set of arguments or parameters by which it can be invoked
- Arguments can be either optional, required or both
- The function returns back only one value
- Combining values in one data structure is a must when the output consists of many different values

Modules in Python

- Developers need to import functions and resources from the modules where they are defined
- There are two techniques to import the module's resources:
  1. Import all resources using the wild card (*)
  2. Import just the needed resources using their names
- To recall a resource or function, it is preferable to qualify it using the module to which it belongs

The `math` Module

Table 2.6: Some of the `math` Module Functions

Function	Description
factorial(*x*)	Find x!. Raise a ValueError if x is negative or non-integral.
ceil(*x*)	Return the ceiling of x as a float.
floor(*x*)	Return the floor of x as a float.
fsum(*iterable*)	Return an accurate floating point sum of values in the iterable.
fabs(*x*)	Return the absolute value of the float x.
log(*x*[, *base*])	Return the logarithm of x to the given base.
pow(*x*, *y*)	Return x**y (x to the power of y).
sqrt(*x*)	Return the square root of x.

References

Lambert, K. A. (2010). Software development, data types, and expressions. In *Fundamentals of Python: From first programs through data structures* (pp.39-73). Boston, MA: Cengage Learning.